# AN INTERNSHIP REPORT ON VIRTUAL MOUSE USING HAND GESTURES PROJECT

**Research Proposal** · March 2024

DOI: 10.13140/RG.2.2.29970.67529

Kamal Acharya
Tribhuvan University
**249** PUBLICATIONS **4,439** CITATIONS

# AN
# INTERNSHIP REPORT

# ON

# VIRTUAL MOUSE USING HAND GESTURES PROJECT

## BY

KAMAL ACHARYA

(Tribhuvan University)

**Date: 2024/03/21**

# Virtual Mouse using Hand Gestures

## Abstract

This paper proposes a virtual mouse system based on HCI using computer vision and hand gestures.Techniques for establishing the process of human-computer interaction(HCI) have evolved since the invention of computer technology. The mouse is its great invention in HCI (Human-Computer Interaction) technology. Wireless or Bluetooth mouse technology is still in development, but the technology is not yet completely device-free. The Bluetooth mouse requires battery power and a connection dongle. The presence of additional devices in the mouse makes it difficult to use. The proposed mouse system solves these limitations. We have written a program for controlling the mouse movement using Python and OpenCV with a real-time camera that detects hand patterns, tracks hand gesture patterns that replaces the work of a traditonal physical mouse. Gestures captured with a integrated camera or webcam are processed with recognition technology. The user can control some of the computer's cursor functions with hand gesturing movements. Primarily, users can left-click, right-click, and double-click by scrolling their hand up or down with various gestures. This system captures frames using a webcam or built-in camera, processes the frames to make them trackable, recognizes various gestures made by the user, and performs mouse functions . Therefore, the proposed mouse system eliminates dependence on device to use the mouse. So the development of HCI technology can help.

## Introduction

The most effective and expressive means of human communication is hand gestures, which is a widely accepted language. It is expressive enough for the deaf and dumb to understand. In this work, a real-time hand gesture system is proposed.
Test setup of the system using the low-cost, fixed-position web camera
mounted on a computer monitor, or a fixed camera on a laptop
,with the system's high-definition recording capability. This image captures a snapshot
using a fixed distance red-cyan [RGB]
color space.

The gesture-controlled virtual mouse simplifies human-computer interaction using hand gestures. There is no need of physical contact between the user and the computer.
All I/O operations can be controlled virtually by static and dynamic hand gestures. This project uses state-of-the-art machine learning and computer
vision algorithms for hand gesture recognition that works seamlessly without any additional hardware requirements. It leverages models like CNN implemented by MediaPipe that run on top of pybind11. It works directly on the hand using MediaPipe hand detection.

This system is implemented in Python programming language using the Computer Vision based library OpenCV. This system has the potential to replace the typical mouse and also the remote controller of machines. The only barrier is the lighting condition. That’s why the system still can’t be enough to replace the traditional mouse as most of the computers are used in poor lighting conditions

## Problem description and Overview

When designing such a system, we require a camera setup that is positioned in a manner so it can see the user’s hands in the right positions clearly that it can track fingertips as a movable object.

## Applications

Video conferencing is very popular these days. For this

reason, most computer users use a webcam on their

computer and most laptops have a built-in web camera. The

the proposed system, which is based on a web camera, is able to

partially eliminate the need for a mouse. Process

interacting with the computer using hand gestures is very

interesting and effective approach to HCI (Human-Computer

Interaction). There is really good research in this area of work. There is also hand gesture recognition technology popular in sign language recognition.

## Objective

The main objective of the proposed virtual AI mouse is to furnish an alternative to the conventional physical mouse that provides mouse functions with the help of computer vision enabled computer that houses a web camera which recognizes fingers and hand gestures and processes the captured frames and uses a machine learning algorithm to execute the defined mouse functions like moving the cursor, right click, left click and scrolling function. Also we are using multiple libraries to perform this project.

## Proposed System

Using the current system, although there are a number of  quick access methods available for hand and mouse gesture for notebooks, we could use laptop or webcam and hand recognition in our project we could use the gesture to control the mouse and perform basic operations such as controlling the mouse pointer, selecting and deselecting using the left button and the quick file access function transmission between systems connected via a LAN cable. The finished project is "zero cost" hand recognition system that uses simple algorithms to do track the hand and hand movements; by assigning an action for each move. But our main focus is primarily on actions such as  pointing and clicking, and also defining an action to transfer files between connected systems using hand movements alone. The system we are implementing is written in much more responsive python code and is easy to implement because python is a simple language, platform independent, flexible and portable; this is what was desired when created a program so focused on that the purpose of was to create a virtual mouse and hand recognition system. The system is much more extensible by defining actions for the movement of the needle to perform a specific action. It can be further modified by performing such actions for the whole hand gesture.

## RELATED WORK

Work on virtual mice has been done in which the user wears gloves for the system to recognize and collect data, and also another system where pieces of colored paper are tied on the hand for gesture recognition. do, although such systems are not feasible to explicitly perform mouse actions. Glove recognition is not always possible and many users do not want to wear gloves or the gloves may not fit properly. In other cases, using colored tips for gesture detection and processing may not always work with low accuracy. Other people have contributed leading up to this system, such as Google's work with MediaPipe (an open source hand detection library).

## ALGORITHM USED FOR HAND DETECTION

In this project work, MediaPipe library, which is an open source cross-platform framework;and the OpenCV library for computer vision are used for hand and finger tracking. This algorithm uses machine learning concepts to track and detect hand and fingertip gestures.

## MEDIAPIPE

The MediaPipe framework is used by developers to build and analyze systems through graphics and it has also been used to develop systems for application purposes. The MediaPipe library is used by developers to design and analyze various models graphically, and many of them have been used to create applications. MediaPipe Hands uses an ML pipeline consisting of multiple models that work together. The MediaPipe embedded model will work in pipeline mode. It mainly consists of graphs, nodes, streams and calculators. The MediaPipe framework is based on three basic parts; it is a benchmark, a framework for retrieving data from sensors and a set of components called computers and they are reusable. A pipeline is a graph made up of components called computers, where each computer is connected by streams through which data packets flow. The sales flow is implemented as a MediaPipe chart using a trailing trailing subgraph from the trailing stops module and displayed using a handgraph renderer. The internal hand signal tracking subgraph uses the hand signal subgraph from the same module and the palm detection subgraph from the palm detection module. Computer and flow combine to create data flow diagrams; image created with MediaPipe where each node is a computer and the nodes are connected by threads. Mediapipe provides cross-platform and customizable open source ML solutions for live and streaming media. This is useful in many situations such as:

1.Selfie segmentation.

2.Face mesh

3.Human pose detection and tracking

4.Holistic tracking

5.3D object detection.

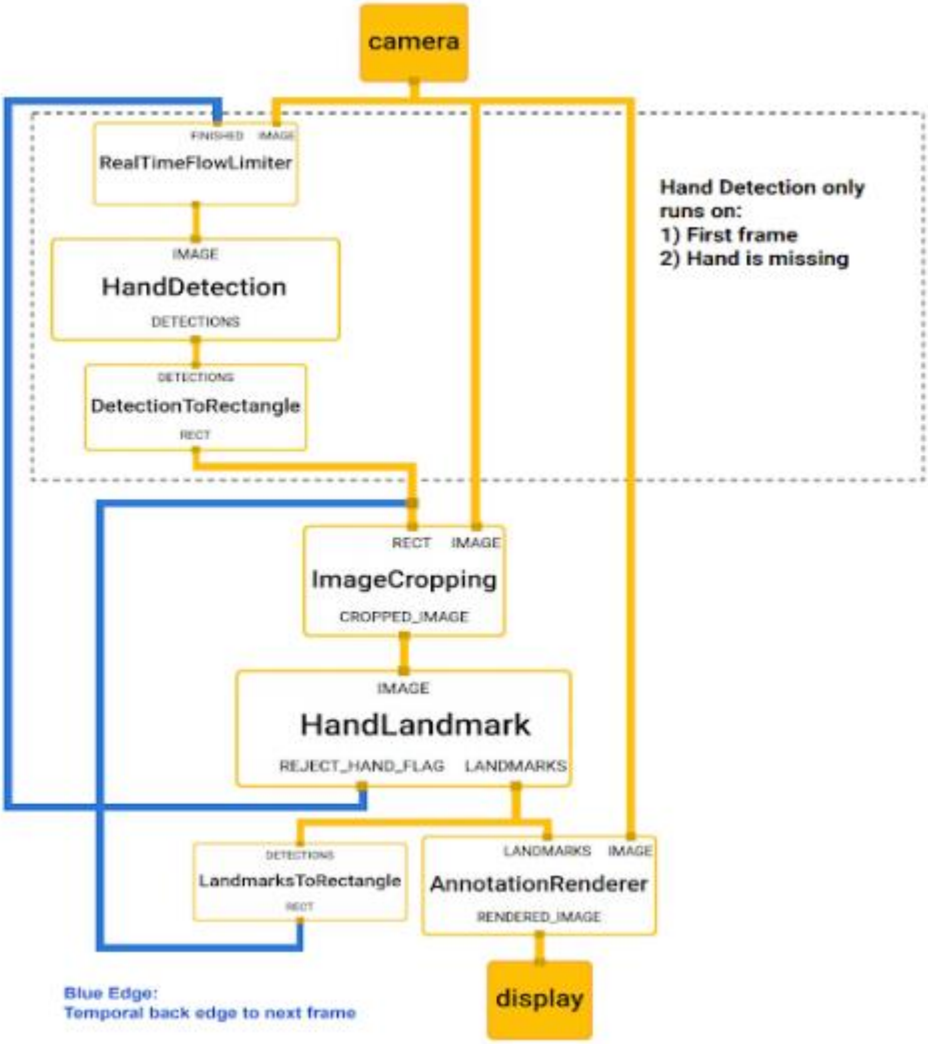


FIG: Hand Recognition graph MediaPipe

## OPENCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. The library contains more than 2500 optimized algorithms, including a comprehensive set of classical and modern machine learning and computer vision algorithms. This library is written in python language and helps to create applications using computer vision. In this model, the OpenCV library is used for image and video processing as well as for face and object detection and analysis. The development of hand gesture recognition in Python and OpenCV can be done by applying the theories of hand segmentation and hand detection system using Haarcascade classifier.

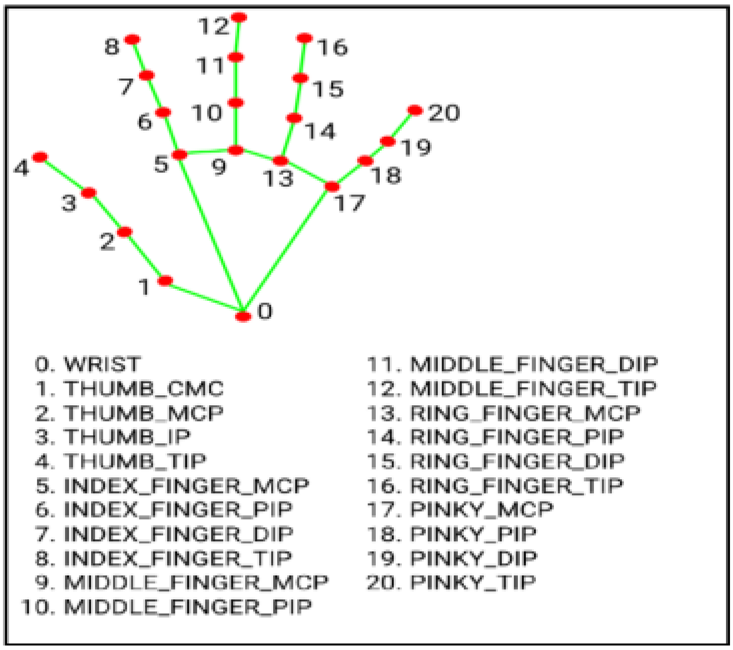


FIG: Hand landmarks points used by MediaPipe

## Methodology

Each component in its working is individually explained in the following subsections respectively:

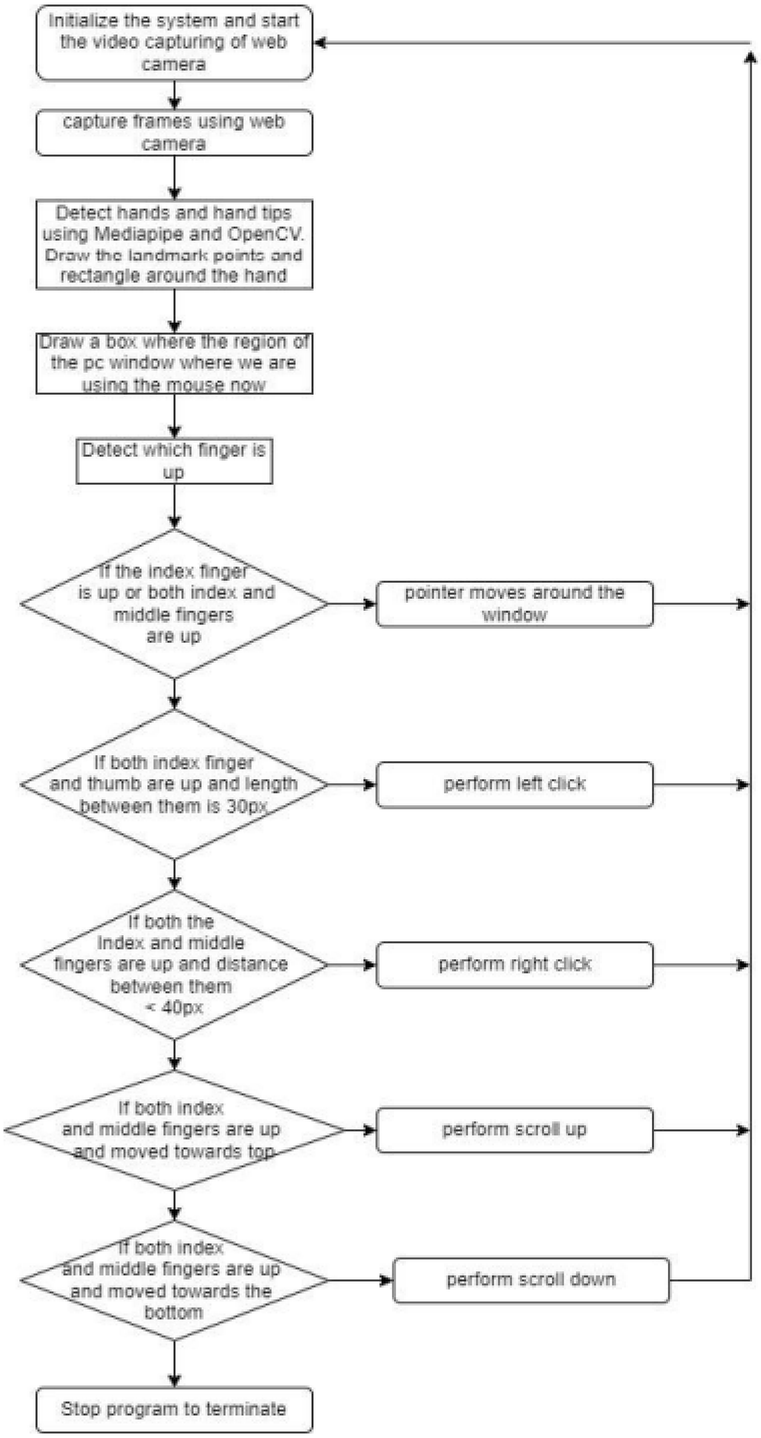
### Image Processing:

### 1.Camera Setup

Runtime operations are handled by the webcam of the connected laptop or desktop. To capture video, we need to create a Video Capture object. Its argument can be the device index or the name of the video file. The device index is just a number to designate which camera. Since we use only one camera, we convert it to '0'. We can add more cameras to the system and stream them as 1,2 and so on.We can then shoot frame by frame.

## 2.Capturing Frames

The infinite loop is used for web camera
to take pictures in each instance and is open during the entire program run. We capture the live stream, frame by frame. We then process each
image captured in the (default) RGB color space to the HSV
color space. There are more than 150
color space conversion methods available in OpenCV. But we will only look at the two most widely used
codes, BGR to Gray and BGR to HSV.


Initialize the system and start the video capturing of web camera
capture frames using web camera
Detect hands and hand tips using Mediapipe and OpenCV. Draw the landmark points and rectangle around the hand
Draw a box where the region of the pc window where we are using the mouse now
Detect which finger is up
If the index finger is up or both index and middle fingers are up
pointer moves around the window
If both index finger and thumb are up and length between them is 30px
perform left click
If both the Index and middle fingers are up and distance between them < 40px
perform right click
If both index and middle fingers are up and moved towards top
perform scroll up
If both index and middle fingers are up and moved towards the bottom
perform scroll down
Stop program to terminate

## PROCESSING THE COLLECTED FRAMES

The web camera continues to collect images until the program below closes. The captured images from the video are collected in BGR color format from the web camera. In order for OpenCV to process images, the BGR color format must be converted to the RGB color format

Subsequently, OpenCV processes the frames to detect hand/s.

```
image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
```

## RECOGNIZING THE GESTURE

At this stage, if the hand is tracked and the finger continues to point, MediaPipe recognizes the finger and tip with the 21 coordinates on the finger, processes the gesture, and performs the corresponding mouse action.

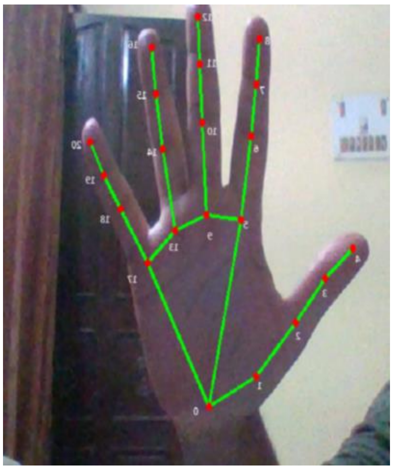


**Fig:Land marks on hand**

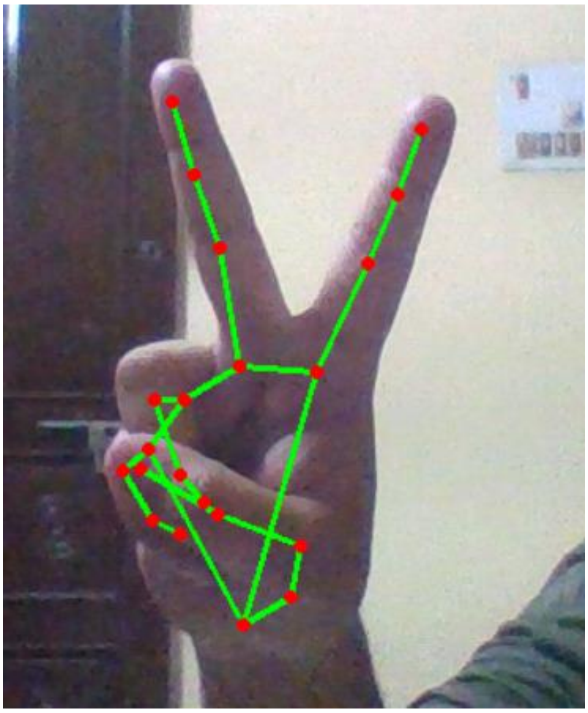


**Fig: moving Cursor with 2 fingers**

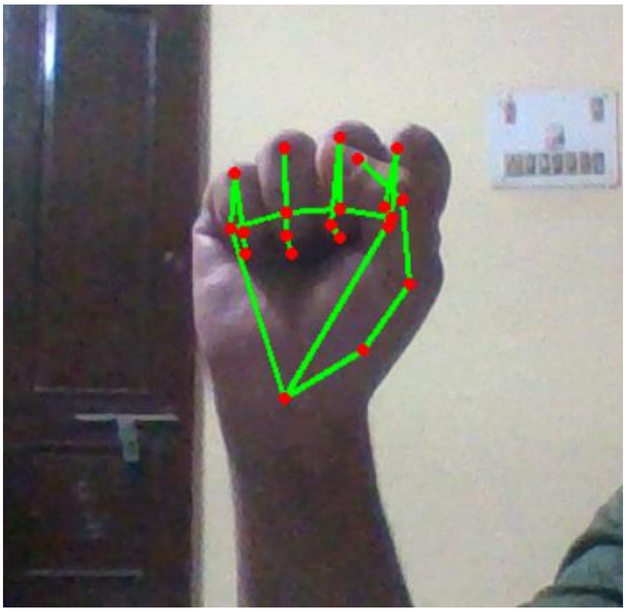


**Fig: Dragging with closed fist**

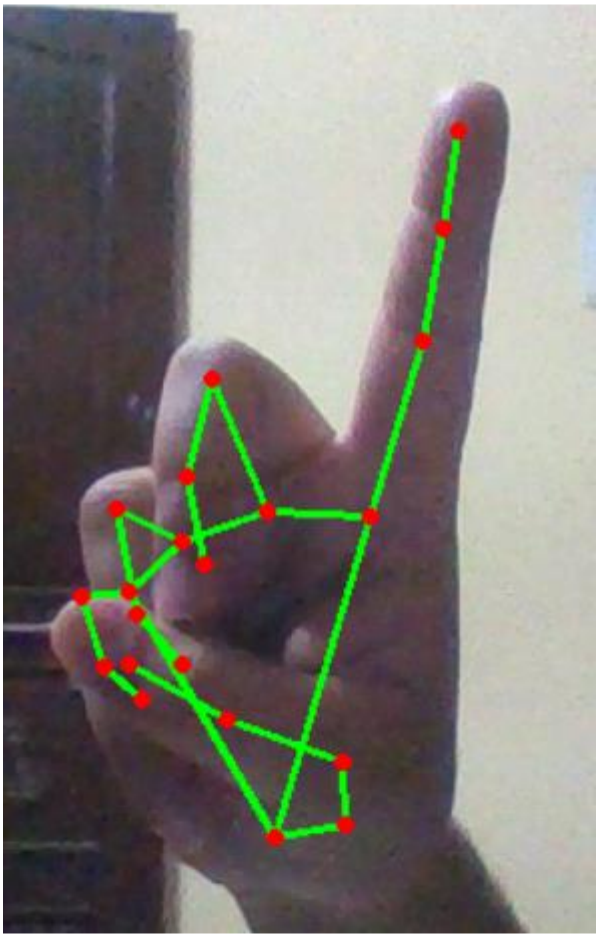


**Fig: Left and right Click with one finger**

## RESULTS AND INFERENCES

Cross-comparing AI virtual mouse system tests is troublesome because only a limited number of datasets can be accessed. Hand gestures and fingertip detection have been tested in a variety of lighting conditions and also tested with different distances from the camera to track and detect hand gestures at your fingertips

The test was performed 40 times by 2 persons resulting in 320 gestures with manual labelling, and this test has been made in different light conditions and at different distances from the screen, and each person tested the AI virtual mouse system 10 times in normal light conditions, 10 times in faint light conditions, 10 times in close distance from the webcam, and 10 times in long distance from the webcam, and the experimental results are tabulated in Table below:

|  |  | Evaluation |  |
|---|---|---|---|
| **Mouse Function** | **Correct operation** | **Incorrect Operations** | **Accuracy** |
| **Pointer Movement** | **75** | **5** | **93.7** |
| **Left click** | **70** | **10** | **87.5** |
| **Right Click** | **72** | **8** | **90.0** |
| **Drag/Drop** | **68** | **12** | **85.0** |
| **Result** | **285** | **35** | **89.06** |

## CONCLUSION

The main objective of the proposed virtual AI mouse is to furnish an alternative to the conventional physical mouse that provides mouse functions with the help of computer vision enabled computer that houses a web camera which recognizes fingers and hand gestures and processes the captured frames and uses a machine learning algorithm to execute the defined mouse functions like moving the cursor, right click, left click and scrolling function.

After testing we have come to the conclusion that the proposed virtual mouse system has worked exceedingly well and with greater accuracy when compared to previously proposed models mentioned in the related work and the current system has overcome the drawbacks of the other systems. As such, this proposed AI based virtual mouse system can be used in real-time and in real-world applications. Additionally, the system eliminates the need to contact high touch surfaces and devices by using hand gestures without using a conventional mouse device.

## References

1. *Kamal Acharya. POST OFFICE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 02, 2024. DOI: https://doi.org/10.22541/au.172261514.44494375/v1*
2. *Kamal Acharya. Fruit shop management system project report. Authorea. August 02, 2024.DOI: https://doi.org/10.22541/au.172261514.42227675/v1*
3. *Kamal Acharya. Dairy management system project report. Authorea. August 02, 2024. DOI: https://doi.org/10.22541/au.172261513.39402347/v1*
4. *Kamal Acharya. DATA COMMUNICATION AND COMPUTER NETWORK MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.37480177/v1*
5. *Kamal Acharya. School management system project report. Authorea. August 01, 2024.DOI: https://doi.org/10.22541/au.172254873.34023165/v1*
6. *Kamal Acharya. A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.30191075/v1*
7. *Kamal Acharya. A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT. Authorea. August 01, 2024 DOI: https://doi.org/10.22541/au.172254872.26972790/v1*
8. *Kamal Acharya. Web chatting application project report management system. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.18588592/v1*
9. *Kamal Acharya. RETAIL STORE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.14590154/v1*
10. *Kamal Acharya. SUPERMARKET MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.19145062/v1*
11. *Kamal Acharya. SOCIAL MEDIA MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.11210579/v1*
12. *Kamal Acharya. Online music portal management system project report. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252488.89734698/v1*
13. *Kamal Acharya. COLLEGE BUS MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245277.70798942/v1*
14. *Kamal Acharya. AUTOMOBILE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245276.67982593/v1*
15. *Kamal Acharya. Ludo management system project report. Authorea. July 31, 2024 DOI: https://doi.org/10.22541/au.172243999.98091616/v1*
16. *Kamal Acharya. Literature online quiz system project report. Authorea. July 31, 2024 DOI: https://doi.org/10.22541/au.172243825.53562953/v1*
17. *Kamal Acharya. Avoid waste management system project. Authorea. July 29, 2024 DOI: https://doi.org/10.22541/au.172228528.85022205/v1*
18. *Kamal Acharya. CHAT APPLICATION THROUGH CLIENT SERVER MANAGEMENT SYSTEM PROJECT. Authorea. July 29, 2024. DOI: https://doi.org/10.22541/au.172228527.74316529/v1*

19. *Kamal Acharya. Parking allotment system project report. Authorea. July 29, 2024. DOI: https://doi.org/10.22541/au.172227078.89966943/v1*
20. *Kamal Acharya. HEALTH INSURANCE CLAIM MANAGEMENT SYSTEM. Authorea. July 26, 2024. DOI: https://doi.org/10.22541/au.172202020.06707762/v1*
21. *Kamal Acharya. ONLINE TRAIN BOOKING SYSTEM PROJECT REPORT. Authorea. July 22, 2024. DOI: https://doi.org/10.22541/au.172167914.45160406/v1*
22. *Kamal Acharya. COVID MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 16, 2024. DOI: https://doi.org/10.22541/au.172116616.60220024/v1*
23. *Kamal Acharya. COVID MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 16, 2024. DOI: https://doi.org/10.22541/au.172116616.60220024/v1*